VISUAL TRACKING USING A MEMORY GUIDED PARTICLE FILTER WITH ANNEALED WEIGHTED QUANTUM-BEHAVED PARTICLE SWARM OPTIMIZATION

By

Saptarshi Sengupta

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

Electrical Engineering

December 16, 2017

Nashville, Tennessee

Approved:

Richard Alan Peters II, Ph.D.

D. Mitchell Wilkes, Ph.D.

To my family in Kolkata, India.
All of this is possible because of their support.

ACKNOWLEDGEMENTS

This work would not have been possible without the financial support by the Department of Electrical Engineering & Computer Science at Vanderbilt University. I am particularly indebted to my research adviser Dr. Richard Alan Peters II for providing useful insights at every stage and constantly guiding me in the right direction. I learnt from Dr. Peters the value of being patient when the going got tough and the sense to nurture yet critique. These are traits I wish to exercise as I go along in life. I am also thankful to Dr. Mitch Wilkes for his inputs on improving the work and to Dr. Bennett Landman for providing a list of suggestions to advance the workflow going forward. I wish to mention INRIA Labs, Grenoble, France for using the CAVIAR Test Suite and would also like to acknowledge the use of Mian's Aircraft Tracking Database.

I want to thank my friends Sanchita, Krishnendu, Peng, Semiu, Daniel and Cole for lending their helping hands whenever need arose. Thanks to Mingqi and Ke for many interesting conversations and teaching me skill transfer experiments between industrial robots. A special note of thanks to Dr. Shyamali Mukherjee whose welcoming approach made my life much easier in Nashville.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

The need for stochastic approaches in visual object tracking

Visual Object Tracking is an active research area within the Computer Vision community and has been rigorously studied due to its relevance in achieving key practical functionalities in today's increasingly complex cyber-physical world. Some of the more well-known applications include real-time video surveillance and security systems, smart traffic monitoring and autonomous vehicle navigation. While object trackers aim to identify distinguishing features of targets across multiple frames of interest in sequential images, several challenging issues arise that pose as potential failure modes. Varying environmental and behavioural conditions such as complex object motion, partial or complete occlusion of the region of interest, changes in illumination and scale, injection of noise etc. lead to inefficient and at many times failed tracking. Constrained optimization approaches in mitigating tracking failures have demonstrated notable success. The existing methods use either deterministic [1-3] or stochastic approaches [4-11]. Deterministic approaches typically employ gradient descent search in order to minimize a cost function and obtain parametric estimates. One such example that has been extensively used is the Snakes model introduced by Kass et al [1]. Hager and Belhumer defined the cost function as the sum of squared deviations of candidate solutions from the ground truth [2] whereas Comaniciu minimized the cost difference between two colour histograms by using the Mean Shift Algorithm [3]. Deterministic approaches are computationally less expensive, however they are susceptible to getting trapped in

local optima. Stochastic approaches involve probabilistic operators and better estimate parameters by intelligently querying the multidimensional search space for the global optima, with the tradeoff being computational load. Several approaches have been proposed in [4-11] which effect better performance compared to their deterministic counterparts but the curse of dimensionality remains for high dimensional problems. Due to the dynamic nature of the environment, a unified object tracking scheme is very difficult to accomplish. Particle Filters [12] are recursive implementations of Monte Carlo methods and are ideal for analyzing highly non-linear, non-Gaussian state estimation problems where classical Kalman Filter based approaches fail. The generic Particle Filter suffers from the degeneracy and Sampling Importance Resampling (SIR) induced particle impoverishment problem, leading to proposed enhancements in the sampling stage as in [13-14].

Overview of the work

This work incorporates a memory guided motion model and a hybrid Quantum-behaved Particle Swarm Optimization (QPSO) resampling scheme using annealing and weighted mean best operators (Annealed Weighted QPSO-AWQPSO) to effectively recast particles to the higher likelihood regions in the posterior probability landscape. The methodology is tested out on two benchmark problems containing a set of environmental test conditions. Performance metrics like Root Mean Square Error (RMSE), Number of Frames Successfully Tracked, Tracking Precision versus Centre Error Threshold, Recall versus Overlap Threshold and Frames per Second (FPS) are analyzed over batches of computations. Such statistical analyses suggest performance improvements using the proposed method in comparison to the Particle Swarm Optimization Resampling inspired Particle Filter (PSO-PF) as well as the standard Particle Filter (PF).

The rest of the work is summarized as follows: Chapter II reviews related work at the intersection of Evolutionary Computation and particle resampling in Particle Filters, Chapter III outlines the resampling techniques used and Chapter IV details the proposed approach. Chapter V lists the tracking quality indices used in the model followed by Chapter VI which elaborates on the experimental conditions and results on benchmark problems. Chapter VII provides an analysis of the results obtained and Chapter VIII concludes the paper with possible directions for future work.

CHAPTER II

RELATED WORK

A Bayesian Inference approach to object tracking

A Bayesian inference approach to the object tracking problem involves dynamic state transition through time using a System Model and state measurement through an Observation Model. A Markovian system model in this regard can be formulated as a state transition from the previous one to the current.

$$X_k = f(X_{k-1}, v_k) \leftrightarrow p(X_k \,|X_{k-1}) \tag{1}$$

The observation model can be expressed as:

$$O_k = h(X_k, \eta_k) \leftrightarrow p(O_k \mid X_k) \tag{2}$$

The sequences $\{X_k\,,\, k \in I^+\}$ along with $\{O_k\,,\, k \in I^+\}$ denote the target states and the measurement set of the state sequence in frame k. v_k and η_k are mutually independent system noise and measurement noise. The central goal of a particle filter is to find an approximation of the posterior probability distribution of observations given the current state $p(O_k \,|X_k)$, using a set of weighted samples drawn from a proposal distribution with an associated particle rank defined by a one to one correspondence between high posterior likelihood and large weight. The weights κ are generally computed using the following proportionality relation:

$$\kappa_k^i \propto \kappa_{k-1}^i \frac{p(O_k \mid X_k^i) p(X_k^i \mid X_{k-1}^i)}{p(X_k \mid X_{k-1}^i, O_k)} \quad (3)$$

which is the previous weight scaled by the ratio of the conditional joint probability of the observation O_k in the current state and the probability of the current state given the previous to the probability of the current state given the joint probability of the previous state and the current observation.

The posterior distribution is then updated as the following weighted sum:

$$p(X_k \mid O_k) = \sum_{i=1}^{N} \kappa_k^i \, \delta(X_k - X_k^i) \quad (4)$$

where $p(O_k|X_k)$ is the likelihood and $\delta(.)$ is the Dirac-delta function.

Sample degeneracy and sample impoverishment

It is fundamentally important to generate a proposal distribution such that the sampled particles belong to the region of significant likelihood of the posterior. Particle filters often run into sample degeneracy problems [15] because a large fraction of particles have negligibly small weights after only a few iterations, Sampling Importance Resampling (SIR), which is a probabilistic particle selection method mitigates the problem somewhat and has therefore been adopted widely as a solution [16]. In the resampling step, particles having small weights have low chances of being propagated to the next iteration. A major weakness of PF-SIR in effectively addressing the *Sample Degeneracy Problem* is that particle diversity can decrease over the course of iterations. This leads to the *Sample Impoverishment Problem* [17] wherein the resampled particle set does not accurately

reflect the underlying statistical properties of the original particle set. As the number of effective particles decreases, the collective information carried by them also declines resulting in suboptimal object representations. The number of effective particles N_{eff} can be expressed as:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N}(\kappa_k^i)^2} \tag{5}$$

Mitigation strategies

The *Sample Impoverishment Problem* has led to several mitigation strategies use prior knowledge processing or multi-layered sampling. Partitioned Sampling [17], Annealed Importance Sampling [19] and Kernel Particle Filters [13] are some of the commonly used techniques in this regard. The Auxiliary Particle Filter by *Pitt* and *Shephard*, 1999 [20] samples particles corresponding to points mapped to an importance density with high conditional likelihood. Some researchers have proposed moving particles of lower importance towards regions of higher posterior likelihood where they might better approximate the underlying probability density function. For example, the Kernel Particle Filter accomplishes this particular objective, however its use of a deterministic search over a continuous probability distribution limits its utility.

In recent years, the use of Particle Swarm Optimization [21] in non-differentiable and ill-structured multidimensional problems has gained popularity due to co-operative exchange of social and cognitive information among swarm members and the relatively low cost of individual particle fitness computation. While it yields promising results for non-differentiable cost functions, it is also limited in its ability to converge to the global best (Van den Bergh, 2001) [22] as per the convergence criteria put forward by Solis and Wet [23]. Numerous updates to the canonical PSO

put forward by Clerc and Kennedy [24], have been made possible by factoring in different initialization conditions, position and velocity updates and hybridization [24-27]. Among these, Quantum-behaved Particle Swarm Optimization (QPSO) [25-26] is a particularly attractive choice as its convergence to an optimum is theoretically guaranteed [27]. Promising results using QPSO-inspired Particle Filters in several tracking datasets have been reported by Sun et al (2015) [7] and by Hu, Fang and Ding (2016) [8].

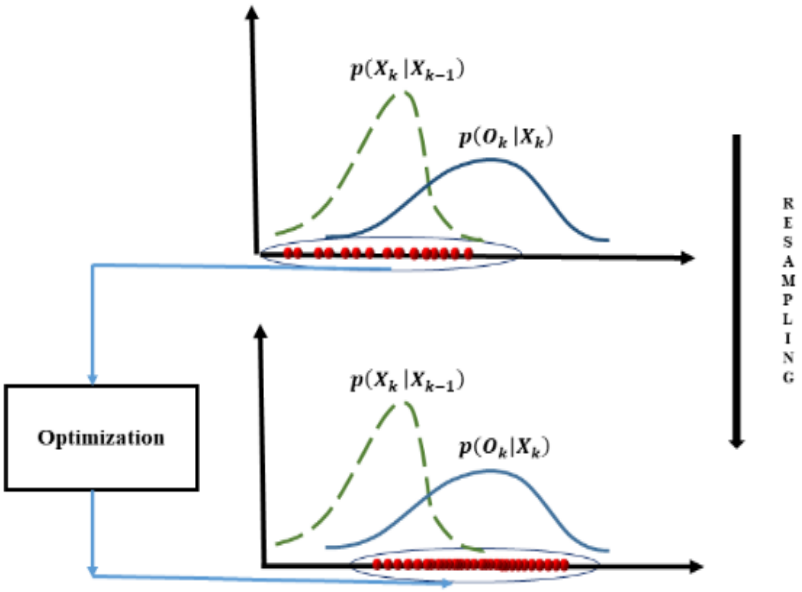


Fig. 1. Particle redistribution towards regions of high likelihood.

CHAPTER III

OUTLINE OF METAHEURISTICS USED

Particle Swarm Optimization (PSO)

PSO [21] is one of many nature-inspired metaheuristics in the broad category of Swarm Intelligence. It has been motivated by observed social co-operation among bird flocks and fish schools. Each particle in PSO is a candidate solution representing a point in a *d-dimensional* search space. In a multidimensional search space, the particles mimic the behaviour exhibited by a group of birds or a school of fish flocking in a multidimensional search space by updating their position coordinates and velocity using information of personal best position so far (*cognitive operator - pbest)* and global best (*social operator - gbest*). An iterative process of movement dependent on social co-operation guides the swarm towards the global optimum. The position and velocity equations in basic PSO are as follows:

$$v_i^{t+1} = \omega\, v_i^t + C_1\, r_1\left(pBest_i - X_i^t\right) + C_2 r_2\left(gBest - X_i^t\right) \tag{6}$$

$$X_i^{t+1} = X_i^t + v_i^{t+1} \tag{7}$$

C_1 and C_2 are cognition and social acceleration constants and r_1 and r_2 are random numbers between 0 and 1 drawn from a uniform distribution. X_i^{t+1}, v_i^{t+1} represent the position and velocity of the i_{th} *d*-dimensional particle respectively at the end of the *t*-th iteration whereas $pBest$ and $gBest$ are the

personal and global best positions. Term 1 in the R.H.S of eq. (6) represents inertia of the swarm and can be adjusted by tuning ω while the next two terms perturb noise in the direction of the individual and population best. The fitness f is updated in the following manner for a cost minimization objective:

$$f(x_i^t) < f(pBest_i)) \Rightarrow pBest_i = x_i^t \tag{8}$$

$$f(x_i^t) \geq f(pBest_i)) \Rightarrow \ pBest_i = pBest_i \tag{9}$$

Algorithm 1 Particle Swarm Optimization

1: for each particle x_i
2: initialize position and velocity
3: end for
4: do
5: for each particle x_i
6: Calculate particle fitness f_i
7: if f_i is better than individual best (pBest)
8: Set f_i as the new pBest
9: end if
10: end for
11: Set best among pBest as the global best (gBest)
12: for each particle
13: Calc. particle velocity acc. to eq. (6)
14: Update particle position acc. to eq. (7)
15: end
16: while max. iter or convergence criterion not met

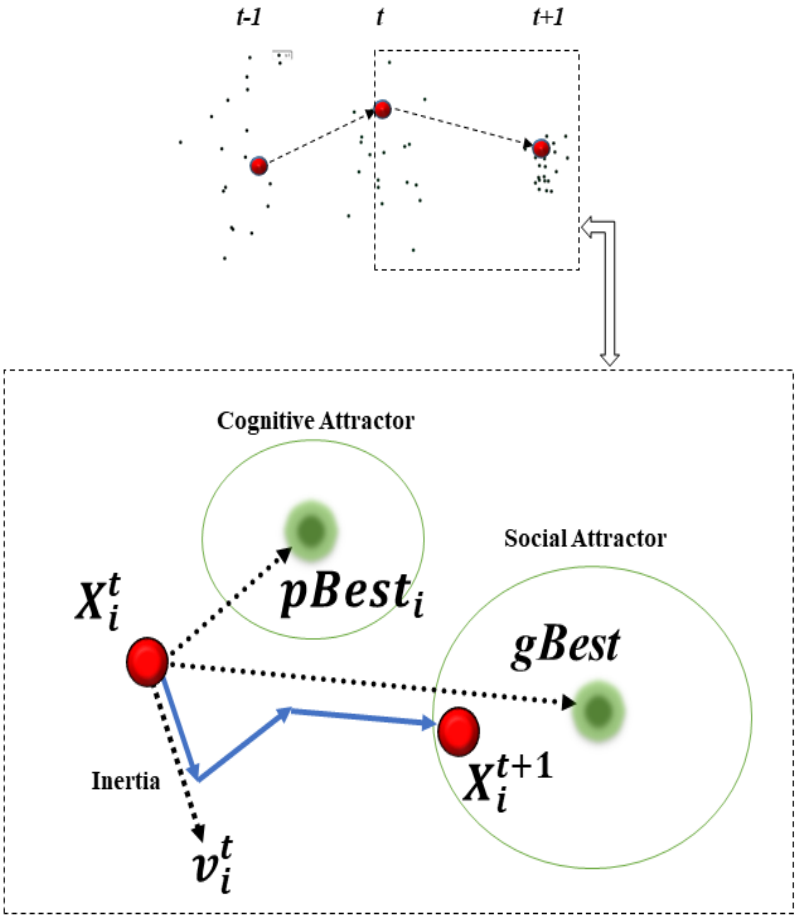


Fig. 2. Particle movement mechanics using PSO.

Quantum-behaved Particle Swarm Optimization (QPSO)

Trajectory Analysis in [28] proved the convergence of PSO necessitates the convergence of each particle to its local attractor $p_i^t = (p_{i1}^t, p_{i2}^t, p_{i3}^t, \dots p_{id}^t)$ and in the process the current position (X_i^t) , the personal best (*pBest*) and the global best (*gBest*) approach the same value. In Quantum-behaved Particle Swarm Optimization, the state of a particle is formally characterized by a wave function ψ with $|\psi|^2$ representing its probability density function. Using recursive Monte Carlo, the QPSO position update equation reduces to:

$$X_{ij}^{t+1} = p_{ij}^t \pm \left(\frac{L_{ij}^t}{2}\right) ln\left(\frac{1}{u_{ij}^t}\right) \tag{10}$$

$u_{ij}^t \sim U(0,1)$ is a uniformly distributed random number and the local attractor p_{ij}^t can be formulated as:

$$p_{ij}^t = \frac{C_1 rand(0,1)_{ij}^t pbest_{ij}^t + C_2 rand(0,1)_{ij}^t gBest_{ij}^t}{C_1 rand(0,1)_{ij}^t + C_2 rand(0,1)_{ij}^t} \tag{11}$$

rand(0,1) generates different random numbers for pairing with cognitive and social operators. Further simplification results in the following widely used form:

$$p_{ij}^t = \Phi_{ij}^t pBest_{ij}^t + (1 - \Phi_{ij}^t) gBest_{ij}^t \tag{12}$$

where $\Phi_{ij}^t \sim U(0,1)$ is a generated random number distributed uniformly.

The parameter $L_{ij}^{t}$is the characteristic length of the underlying wave function and is evaluated as:

$$L_{ij}^{t} = 2\beta \left| p_i^{t} - X_{ij}^{t} \right| \tag{13}$$

The contraction-expansion co-efficient β is tuned to maintain the balance between exploration and exploitation. The complete position update equation is thus given by:

$$X_{ij}^{t+1} = p_{ij}^{t} \pm \beta \left| p_i^{t} - X_{ij}^{t} \right| ln\left(\frac{1}{u_{ij}^{t}}\right) \tag{14}$$

L_{ij}^{t} controls the accuracy and convergence speed of QPSO. The *"Mainstream Thought"* or *Mean Best*, introduced in [25] is the mean of all $pBest$ positions of the particles.

$$mbest^{t} = (mbest_1^{t}, mbest_2^{t}, \dots, mbest_d^{t}) \tag{15}$$

$$= \left[\frac{1}{M}\sum_{i=1}^{M} p_{i1}^{t}, \frac{1}{M}\sum_{i=1}^{M} p_{i2}^{t}, \dots, \frac{1}{M}\sum_{i=1}^{M} p_{id}^{t}\right]$$

An alternate way of writing the position update equation is adopted by re-expressing L_{ij}^{t} :

$$L_{ij}^{t} = 2\beta \left| mbest_j^{t} - X_{ij}^{t} \right| \tag{16}$$

This yields the final form of the popular mainstream thought based position update equation of the QPSO algorithm.

$$X_{ij}^{t+1} = p_{ij}^{t} \pm \beta \left| mbest_{j}^{t} - X_{ij}^{t} \right| ln\left(\frac{1}{u_{ij}^{t}}\right) \quad (17)$$

The second term in the RHS of (17) is additive when a generated random number is less than 0.5 and is subtracted when u^t_{ij} is greater than 0.5.

Algorithm 2 Quantum-behaved PSO

1: for each particle xi
2: initialize position
3: end for
4: do
5: Compute mean best position using eq. (15)
6: for each particle xi
7: for each dimension j
8: Calculate local attractor using eq. (12)
9: if rand(0,1)<0.5
10: Update pos. using eq. (15) with '+'
11: else Update pos. using eq. (15) with '-'
12: end if
13: end for
14: Evaluate fitness function
15: Update pBest according to eq. (8) and (9)
16: end for
17: Set best among pBest as the global best (gBest)
18: while max. iter or convergence criterion not met

CHAPTER IV

ANNEALED-WEIGHTED QPSO FOR VISUAL OBJECT TRACKING

Particle propagation

The uniform weighting scheme in the *Mean Best* calculation in eq. (15) is not an optimum choice as particles of varying fitness values contribute equally to it. Thus, in alignment with predator-prey population models where the fitter of the two survives to pass on their genes, the mean best update is recomputed by assigning a set of variable weights with the particles. Each particle is associated with a weight in proportion to its fitness value thereby making it favorable for the fittest particle to contribute most to the mean best update [27].

The *mbest* calculation thus changes to:

$$mbest^t = (mbest_1^t,\ mbest_2^t, \ldots, mbest_d^t) \tag{18}$$

$$= \left[\frac{1}{M}\sum_{i=1}^{M} \tau_{i1}^t p_{i1}^t, \frac{1}{M}\sum_{i=1}^{M} \tau_{i2}^t p_{i2}^t, \ldots, \frac{1}{M}\sum_{i=1}^{M} \tau_{id}^t p_{id}^t\right]$$

where τ_{ij}^t is the *j*-th dimensional weight of the *i*-th particle in iteration *t*. The standard QPSO suffers from unsatisfactory fine tune during the latter part of the search process [29] and the fitness update scheme rejects particles whose likelihood values are worse than the personal best. However, these particles may evolve over iterations to guide the swarm towards the globally optimum mode and disregarding them from the start of the search process may effectively reduce the diversity of the

swarm. Thus, the fitness update scheme is replaced by an exponential acceptance score where the probability of accepting a particular particle is given by the Metropolis criterion [30]:

$$\theta = \begin{cases} 1, \ if \ \Delta f < 0 \\ exp\left(-\frac{\Delta f}{T_t}\right), \ otherwise \end{cases} \tag{19}$$

Δf is the difference in fitness, θ is probability that the current particle is accepted and T_t is the annealing temperature in iteration t. A suitable exponential cooling schedule is adopted with an initial high value of T_0:

$$T_t = T_0 exp(-t) \tag{20}$$

The value of the contraction-expansion factor β is decreased linearly from 0.9 to 0.5 over the iteration count to facilitate exploitation in the latter part of the search:

$$\beta = (0.9 - 0.5)\left[\frac{(t_{max} - t_{current})}{t_{max}}\right] + 0.5 \tag{21}$$

Algorithm 3 Annealed Weighted QPSO

for each particle x_i
    initialize position
end
do
  Compute mean best position using eq. (18)
  for each particle x_i
    for each dimension j
      Calculate local attractor using eq. (12)
      if rand(0,1)<0.5
        Update pos. using eq. (17) with '+'
      else Update pos. using eq.(17) with '-'
      end if
    end for
  Accept new solution according to eq. (19)
  Update pBest according to. eq. (8) and (9)
  end for
  Set best among pBest as the global best (gBest)
while max. iter or convergence criterion not met

Proposal distribution and sensing model

The dynamic state update stage of the filter makes use of a weight normalized velocity looking back three steps in memory. A Gaussian distribution $X_{k+1} \sim N(X_k, \Sigma_M)$ is used to spread particles

around the current state which results in the following motion model with the importance weight vector λ sorted in ascending order of values. Σ_M is the covariance matrix of the distribution, v_f is the adaptive step size update, Ω is a uniform random number in [-1,1] and v_g is the velocity of the g-th frame.

$$X_{k+1} = X_k + \Omega v_f \tag{22}$$

$$\lambda = sort\left(\left\{\frac{v_g}{\sum_{e=k-2}^{k} v_e}\right\}_{g=k-2}^{k}, descending\right) \tag{23}$$

$$v_f = 2\sum_{a=0}^{2} \lambda\{a+1\}.\{v_{k-a}\} \tag{24}$$

Now, it is known to all that a good observation model is critical to implementing an efficient tracker. However, in practice varying conditions necessitate the use of specific feature descriptors for different tracking scenarios. In this work, the appearance of the targeted object is modeled using a Gaussian fitness function as:

$$f(C,\Sigma) = \left(\frac{1}{2\pi^{n/2}|\Sigma|^{1/2}}\right)\exp\left(-\frac{\Delta^2}{2}\right) \tag{25}$$

$\Delta = \sqrt{(C - C_{GT})^T \Sigma^{-1}(C - C_{GT})}$ is the *Mahalanobis distance* of the observable C with respect to the goal state C_{GT} given covariance Σ. Here, colour cue is used as the feature descriptor to construct likelihood scores because of its simplicity in implementation while providing invariance to translational and rotational change, as well as scale change and partial occlusion. The Euclidean

distance between *i*-th of *N* particles and the manually annotated ground truth for the *k*-th frame is used in subsequent centre error estimation and is given by the following equation:

$$d_i^k = \sqrt{(X_{GT} - X_i^k)^2} \quad \forall i \; in \; I^+ \in [1, N] \tag{26}$$

Proposed tracker model

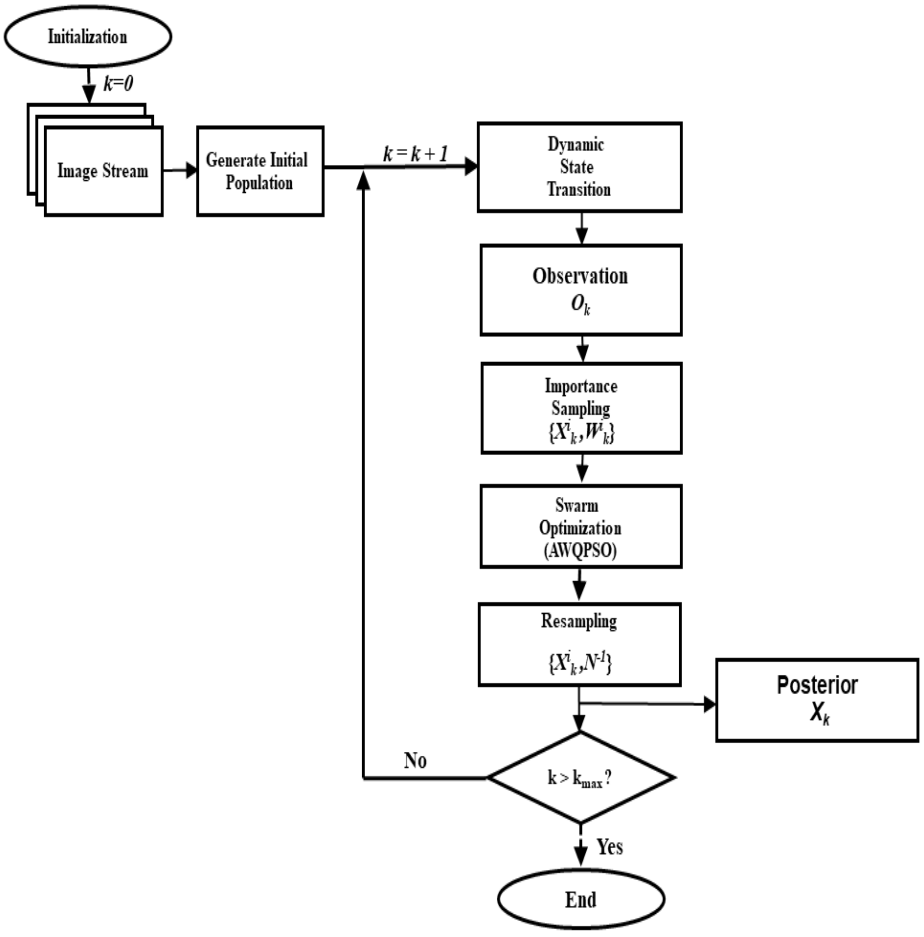


Fig. 3. Flowchart of the AWQPSO tracking model.

CHAPTER V

TRACKING PERFORMANCE INDICES

A quantitative characterization of tracker performance has been made using precision and recall evaluated over the test sequences. Precision, in the context of visual tracking can be defined as the ratio of the number of frames over the total having a centre to swarm deviation less than a preset threshold. Recall, on the other hand is the ratio of number of frames over the total that pass a tracker to ground truth bounding box overlap score greater than a preset threshold. In more formal terms, these are expressed as:

Precision:

$$Precision = \frac{Frames_{RMSE<Threshold}}{Total\ Frames} = \frac{TP}{TP+FP} \tag{27}$$

Recall:

$$Recall = \frac{Frames_{Overlap\ Score>Threshold}}{Total\ Frames} = \frac{TP}{TP+FN} \tag{28}$$

Overlap Score:

The Overlap Score is computed as $\left(\frac{BB_{Tracker} \cap BB_{Ground\ Truth}}{BB_{Tracker} \cup BB_{Ground\ Truth}}\right)$. TP, FP and FN are true positives, false positives and false negatives, respectively and BB denotes the Bounding Box.

Root Mean Square Error:

$$RMSE = \frac{\sum_{k=1}^{k_{max}} \left[\frac{\sqrt{\sum_{z=1}^{N} \left\{ (X_{z,x} - X_{GT,x})_k^2 + (X_{z,y} - X_{GT,y})_k^2 \right\}}}{particles} \right]}{k_{max}} \tag{29}$$

Frames per Second:

$$FPS = \frac{Total\ Number\ of\ Frames\ Executed}{Processing\ Time\ Required\ In\ Seconds} \tag{30}$$

CHAPTER VI

EXPERIMENTS AND RESULTS

Experimental setup

To evaluate the performance of the AWQPSO tracker, three competitive tracking algorithms viz. PF (described in Chapter II), PSO – PF (described in Chapter III) and AWQPSO-PF (described in Chapter IV) have been considered. A comparative analysis of computational load and error margins are calculated using the same observation model for all. Two different video sequences acquired at 25 fps are taken. The first one is the dataset *OneStopNoEnter2cor.mpg* from the EC Funded CAVIAR project/IST 2001 37540 [31]. The Corridor Views of the Lisbon Sequence from the CAVIAR Project are considered. These sequences are shot in a shopping mall using a surveillance camera and variations include scale change, different lighting conditions, nearby moving object (particle hijacking problem) and partial occlusion. The second sequence is *aerobatics_1.avi* from the *Aircraft Tracking Database-Open Remote Sensing* [32] which introduces scale change, camera movement, abrupt motion and specular reflection into the observation.

Table 1.

List of Implementation Terms and Parameters for the Metaheuristic Algorithms.

Term	Discussion
	Some General Terms
Population (X)	The collection or *'swarm'* of agents employed in the search space
Fitness Function (f)	A measure of convergence efficiency
Current Iteration	The ongoing iteration among a batch of dependent/independent runs
Maximum Iteration Count	The maximum number of times runs are to be performed

Particle Filter

Population (X)	Collection of agents approximating states of target under consideration
Proposal	Initial guess of possible target states given some/no apriori knowledge
Observation	Sensed states of the target after the prediction stage is complete
Importance Weights (κ)	A high posterior likelihood implies a large weight
Effective Sample Size (ESS)	Low value of Effective Sample Size implies necessity of resampling

Particle Swarm Optimization

Position (X)	Position values of individual swarm members employed in a multidimensional search space
Velocity (v)	Velocity values of individual swarm members
Cognitive Acceleration Co-efficient (C1)	Empirically found scale factor of pBest attractor
Social Acceleration Co-efficient (C2)	Empirically found scale factor of gBest attractor
Personal Best (pBest)	Position corresponding to historically best fitness for a swarm member
Global Best (gBest)	Position corresponding to best fitness over history for swarm members
Inertia Weight Co-efficient (ω)	Facilitates and modulates exploration in the search space
Cognitive Random Perturbation (r_1)	Random noise injector in the Personal Best attractor
Social Random Perturbation (r_2)	Random noise injector in the Global Best attractor

Quantum-behaved Particle Swarm Optimization

Local Attractor	Set of local attractors in all dimensions
Characteristic Length	Measure of scales on which significant variations occur
Contraction-Expansion Parameter (β)	Scale factor influencing the convergence speed of QPSO
Mean Best	Mean of all personal bests across all particles, akin to leader election in the biological world

Annealed Weighted Quantum-behaved Particle Swarm Optimization

Weighted Mean Best	Fitness weighted mean of all personal bests across all particles
Metropolis Criterion	Criterion facilitating inclusion of worse performing particles in the solution pool to preserve diversity of the swarm
Annealing Temperature	Temperature of the system in a particular iteration in the simulated annealing process [33]
Initial Annealing Temperature	Initial temperature of the system in the simulated annealing process
Contraction Expansion Parameter (β)	Linearly decreasing factor influencing convergence speed of QPSO

Parameter settings

The values of the cognitive and social learning constants *C1* and *C2* in Table 1 are both set to 2.05 as these are empirically found to be the optimal pair. The inertial constant ω in PSO is set to 0.5 after testing a linear time varying inertia weight (TVIW) as well as in increments of 0.1 between 0.1 and 0.9 for PSO which results in a fine balance between exploration and exploitation. The contraction-expansion factor β in AWQPSO is reduced linearly with the number of iterations to explore the search space more in initial iterations and hone in on potential solution regions towards the latter iterations. The population size in all test cases are taken to be 300 to allow for reasonably on-target behaviour across all frames for each algorithm, exceeding which the time cost increases with negligible change in the number of off-target frames. A sufficiently large fitness score computed with respect to the goal state or a maximum iteration count of 50 are kept as the termination criterion for all in-frame optimization using the algorithms.

The methodologies discussed so far are implemented on MATLAB R2016a using an Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz with 8GB RAM and the independent performances over 30 trials are analyzed. No use of Graphics Processing Units (GPUs) have been made during the experiments.

Table 2.

Parameter selection for the tracking algorithms.

Parameter	**Population**	**C1**	**C2**	ω	β	t_{max}	T_t
Value	300	2.05	2.05	0.5	$(0.9\text{-}0.5)[(t_{max}\text{-}t_{current})/t_{max}]+0.5$	50	100

Results for Benchmark Problem 1: OneStopNoEnter2cor

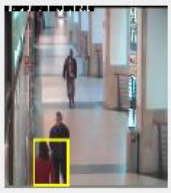
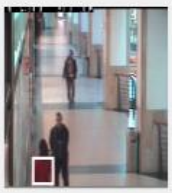
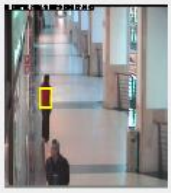
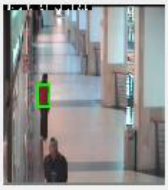
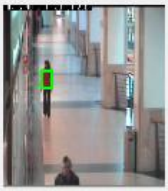
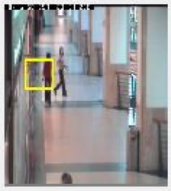
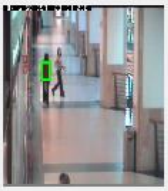
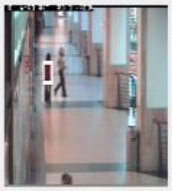
Frame	PF	PSO-PF	AWQPSO-PF
805			
897			
966			
1035			
1081			

Fig. 4-18. Tracking results for OneStopNoEnter2cor. *

* Figures 4-18 should be interpreted in a row-major order.

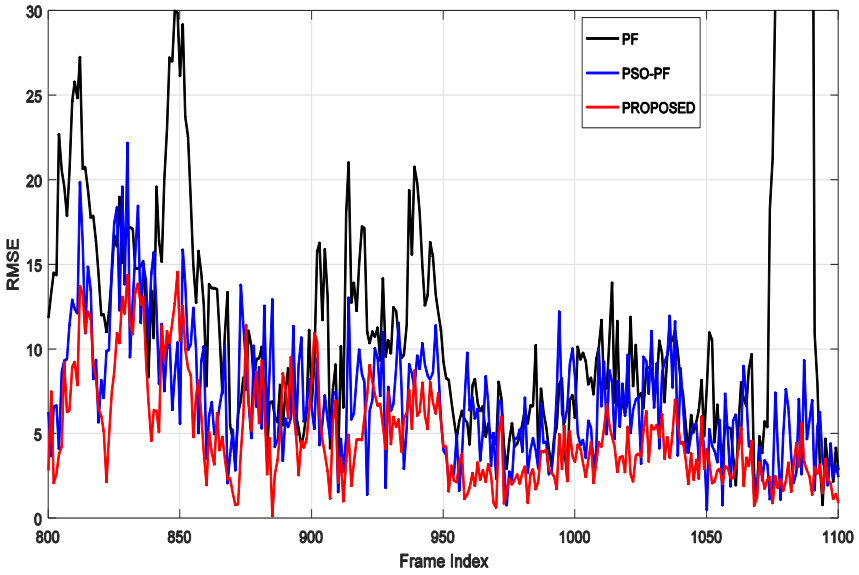


Fig. 19. Evolution of RMSE for OneStopNoEnter2cor (301 frames).

Table 3.

Performance comparison of the three trackers for OneStopNoEnter2cor.

Dataset	Algorithm	FPS	Lost Targets	
			CET=20	CET=30
OneStopNoEnter2cor	PF	17.23±0.3058	40/301	28/301
	PSO-PF	6.71±0.7285	4/301	0/301
	AWQPSO-PF	8.69±0.7044	0/301	0/301

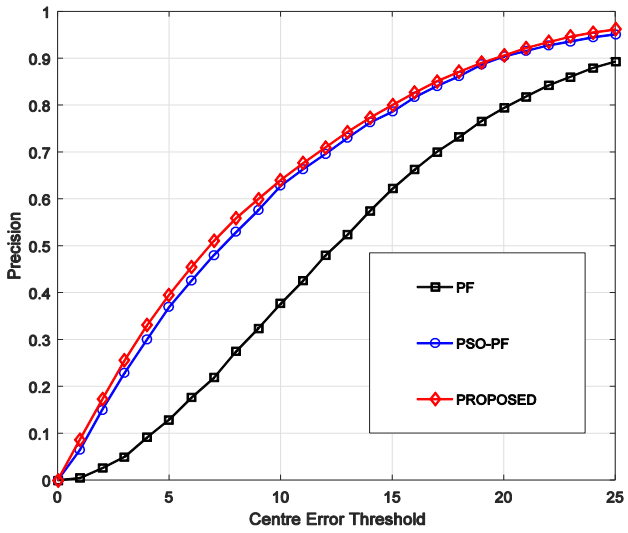


Fig. 20. Precision versus Centre Error Threshold for dataset OneStopNoEnter2cor.

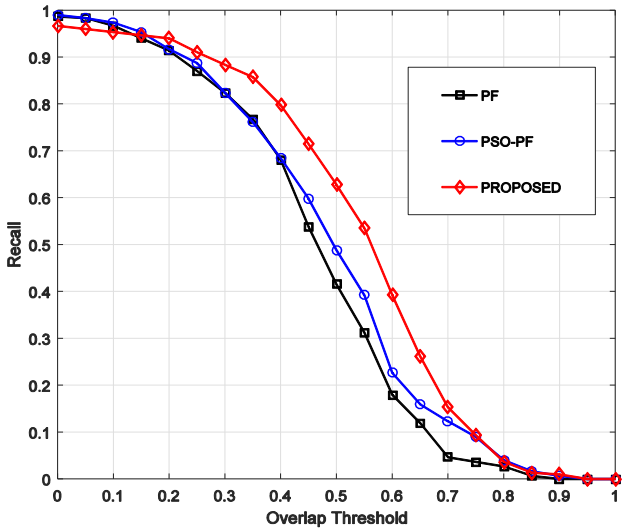


Fig. 21. Recall versus Overlap Threshold for dataset OneStopNoEnter2cor.

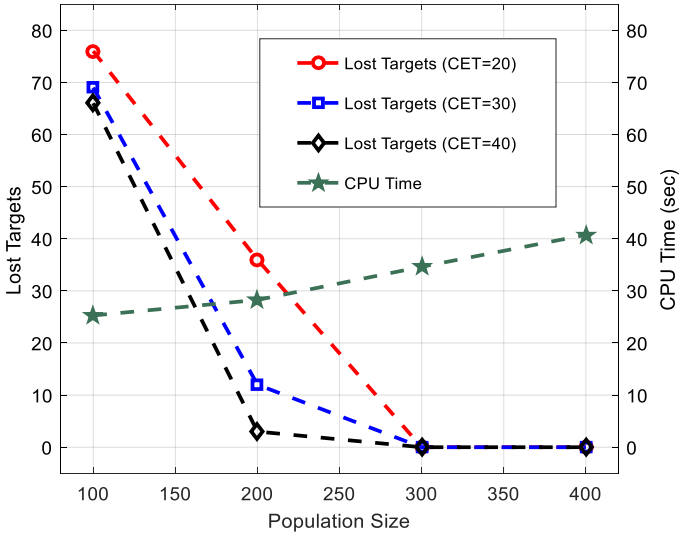


Fig. 22. Performance of AWQPSO under varying population sizes for dataset OneStopNoEnter2cor.

Results for Benchmark Problem 2: aerobatics_1

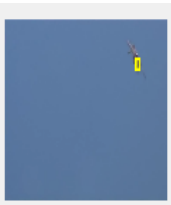
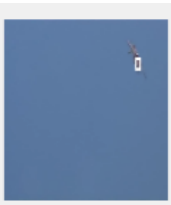
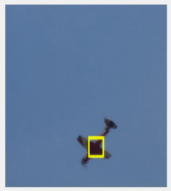
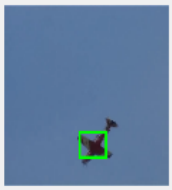
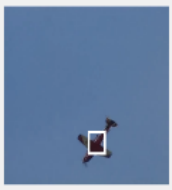
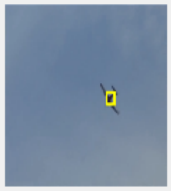
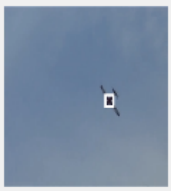
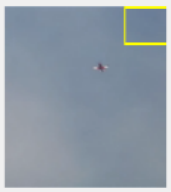
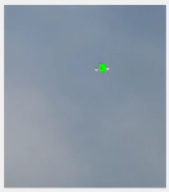
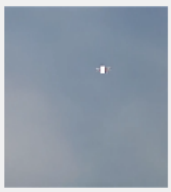
Frame	PF	PSO-PF	AWQPSO-PF
324			
432			
513			
540			
597			

Fig. 23-37. Tracking results for aerobatics_1.*

* Figures 27-41 should be interpreted in a row-major order.

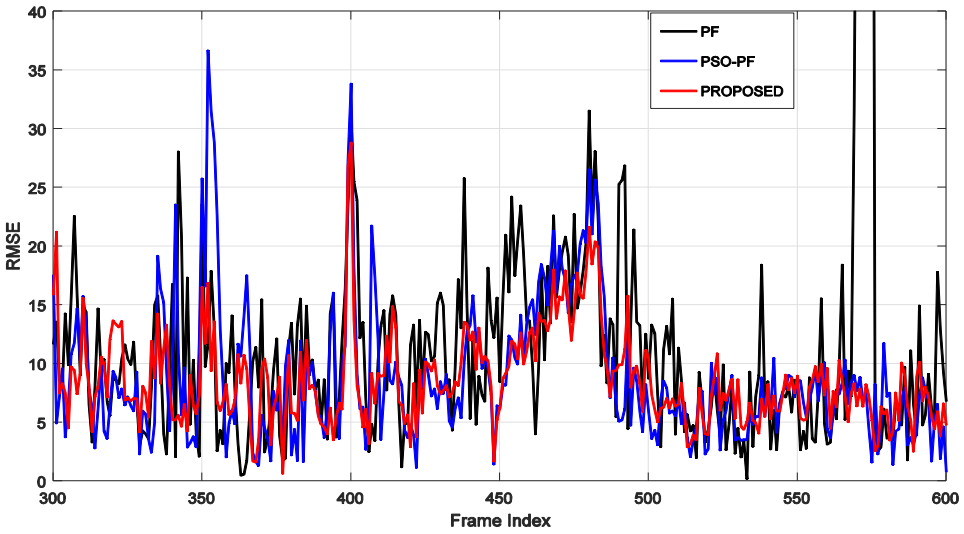


Fig. 38. Evolution of RMSE for aerobatics_1 (301 frames).

Table 4.

Performance comparison of the three trackers for aerobatics_1.

Dataset	Algorithm	FPS	Lost Targets	
			CET=20	CET=30
aerobatics_1	PF	14.34±0.2016	32/301	9/301
	PSO-PF	5.40±0.4783	18/301	3/301
	AWQPSO-PF	5.79±0.3158	5/301	0/301

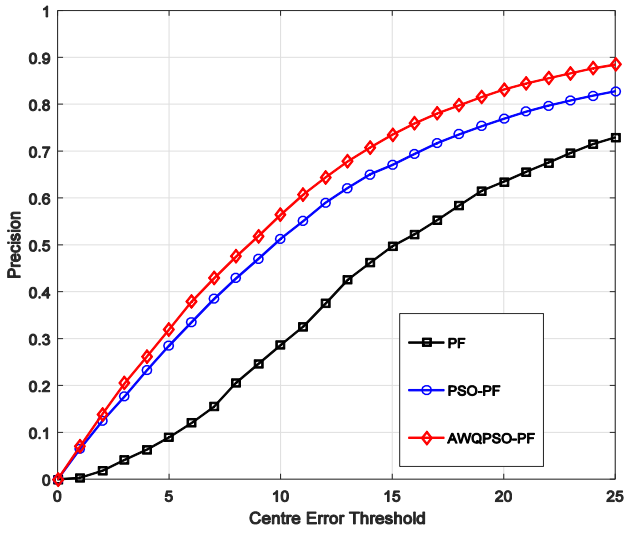


Fig. 39. Precision versus Centre Error Threshold for dataset aerobatics_1.

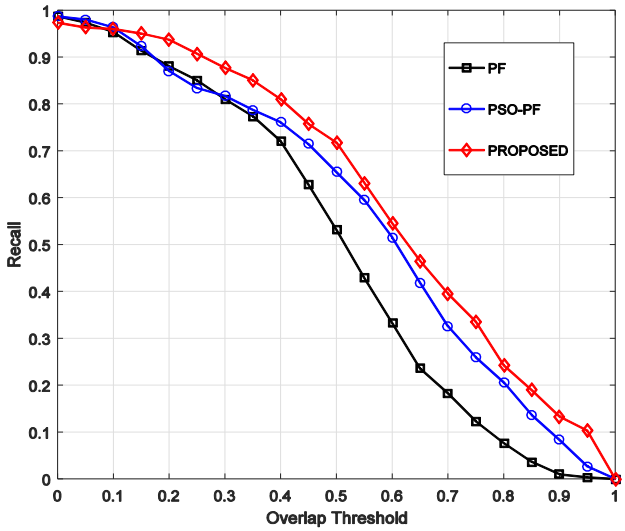


Fig. 40. Recall versus Overlap Threshold for dataset aerobatics_1.

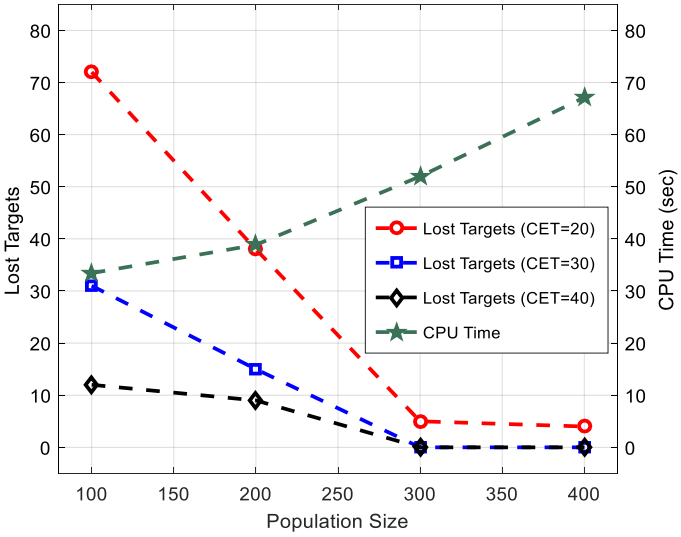


Fig. 41. Performance of AWQPSO under varying population sizes for dataset aerobatics_1.

CHAPTER VII

ANALYSIS OF EXPERIMENTAL RESULTS

Using 300 particles, there is an increase in FPS by 29.51% and 7.22% in case of OneStopNoEnter2cor and aerobatics_1 using AWQPSO over PSO in the Particle Filtering framework. The precision plot for OneStopNoEnter2cor suggests at least 80% of frames pass the RMSE threshold of 15 for both AWQPSO and PSO while that for aerobatics_1 suggests the same percentage of frames pass the RMSE thresholds of 18 and 23 for AWQPSO and PSO. There are 13% and 5% increases in number of frames with a 50% overlap between ground truth and tracker bounding boxes when using AWQPSO as compared to PSO for OneStopNoEnter2cor and aerobatics_1 respectively. In Frames 1075 through 1091 of OneStopNoEnter2cor, the PF tracker is distracted by mistaking local objects as the target, whereas PSO-PF and AWQPSO-PF maintain tracking the target viz. a human subject walking down the corridor clad in red clothing successfully with RMSE<10. Additionally, in aerobatics_1 for Frames 566 to 575 and 594 to 600, PF loses track of the target aircraft due to abrupt motion coupled with scale change, however PSO and AWQPSO trackers perform efficiently. In both the periods though, the proposed AWQPSO-PF tracker has a lower RMSE than the PSO-PF tracker.

Table 3 lists the results of performance parameters for the OneStopNoEnter2cor sequence using the different techniques. Although experimental results suggest that the AWQPSO-PF approach tracks the target with the least net error as compared to PF and PSO-PF, it takes at least twice as much time to process the same number of frames as the Particle Filter does. The number of lost targets for Centre Error Threshold of 20 and 30 are least in AWQPSO-PF and its RMSE is less

than 20 in each of the 301 frames of the subsequence, whereas PF and PSO-PF fail to confine the RMSE to under 20 in all frames. The number of correctly tracked frames (no lost targets) given a RMSE threshold of 20 rose by 1.328% and 13.289% using the proposed approach over PSO and PF respectively. While the AWQPSO-PF and PSO-PF approaches reported same number of correctly tracked frames for RMSE threshold of 30, there was an increase of 9.302% noticed with regard to the PF performance for AWQPSO-PF.

Results from Table 4 indicate AWQPSO-PF has a much tighter bounding box around the target in each frame when compared to the other methods. For instance, the number of frames in the subsequence where the swarm RMSE is less than or equal to 20 is 296 and 283 in case of AWQPSO-PF and PSO-PF respectively – an improvement of 4.318%. Similarly, the concerned number of frames are 298 and 301 for swarm RMSE less than or equal to 30 meaning an improvement of 0.996% using AWQPSO-PF over PSO-PF. The proposed approach reported 8.970% and 2.990% increase in said number of frames for RMSE bounds of 20 and 30 against the standard PF for the AWQPSO-PF tracker.

CHAPTER VIII

GENERAL DISCUSSION

The present study has presented and tested an evolutionary Particle Filter which makes use of an Annealed - Quantum-behaved Particle Swarm Optimization with a weighted Mean Best operator. The better global search ability of the fitness weighted QPSO along with the probabilistic rejection of inferior solutions using Metropolis Criterion makes the proposed metaheuristic well suited for avoiding local minima in the tracking search space. This preserves the diversity of the posterior population and alleviates the sample impoverishment issue to an extent better than the competing Particle Swarm Optimization based Particle Filter and the standard Particle Filter. This is evidenced by the experimental results obtained in Tables 3 and 4 as well as by the indices in Figures 20, 21, 39 and 40. In addition to this, a motion model that looks back three steps in memory is adopted to smooth out sudden changes in velocity of the target. The proposed algorithm is tested using two sequences and is seen to outperform its competitors in both, yielding better RMSE across majority of frames as well as greater area under the curve for both the Precision versus Centre Error Threshold and Recall versus Overlap Threshold metrics. It is observed that the computational load for the AWQPSO-PF method is lower than the PSO-PF, albeit both being significantly slower than the standard PF tracker. This is because of the lesser number of within-frame iterations required by AWQPSO to reach the convergence threshold. However, given the large number of particles used in all the methods and the large within-frame cutoff iteration of 50, the setup is not suitable for real time operation without a reduction in population size and number of in-frame iterations or a parallelized implementation.

The observation model may be modified to accommodate a multi cue likelihood function requiring a multi-objective optimization approach thus effecting a better representation of the target. Additionally, the current AWQPSO-PF tracker model can be extended to track multiple targets with a focus on occlusion handling and evasion of stagnation in local minima over a large number of datasets. Importantly enough, the speedup through parallel computation of particle trajectories in the dynamic state transition section and the subsequent metaheuristic optimization module may lead to a significant increase in FPS. As with existing swarm optimization inspired tracking models such as the Cuckoo Search inspired PF tracker in [9], the QPSO-PF tracker in [7], the Cellular QPSO-PF tracker in [8] and other recent ones [10-11], the current metaheuristic too is susceptible to performance degradation due to incorrect parametric tuning, necessitating a thorough characterization of the operating ranges of its system variables to guarantee convergent behaviour.

Appendix A.

A BASIC PARTICLE FILTERING ALGORITHM

The particle filter is a Monte Carlo approximation technique which implies that the posterior distribution $p\ (X|O)$ is expressed as a collection of samples, also known as particles. A particle filtering model is set up using:

***X**: State Variables*

***O**: Measurements/Observation*

***υ**: Dynamic Noise*

***η**: Measurement Noise*

***f**: State Transition Equation*

***h**: Observation Equation*

The state transition equation and observation equations can be discontinuos, non-linear and non-differentiable and the Dynamic and Observation Noise can be non-Gausssian as long as it is tractable.

$X=(x,\kappa)$ where x are the samples and κ are the associated weights. The number of particles chosen to represent the posterior density $p(X|O)$ is sometimes referred to as the fidelity of the posterior and is a function of the complexity of the posterior and the number of problem dimensions.

The predict-update cycle of the Particle Filter is given below:

Algorithm 4 The Particle Filter

1: for each particle x_i

2: initialize state

3: end for

4: do

5: for each particle x_i

6: Propagate through state transition equation

7: Update weight vector using new observation O

8: Normalize updated weights to sum to one

9: Compute the desired output as an expectation of updated position and weight

10: end for

11: Check if Effective Sample Size (ESS) $< \zeta$ *where* $\zeta \epsilon [0,1]$ *and if so resample*

12: Increase iteration count

In effect, the particles cluster to new locations in each iteration and new weights are updated depending on how well the proposed transitions match the observations. The weights are renormalized to make their sum equal to one so that they represent a probability distribution.

Appendix B.

RESAMPLING IN PARTICLE FILTERS

After the proposal distribution has been updated in an iteration, some particles are prone to veering away towards positions of low likelihood, thus reducing their weights to values close to zero. The decision to resample from the population can be taken after calculating the co-efficient of variation. Formally, the co-efficient of variation (CV) may be expressed as:

$$CV = \frac{var(\kappa)}{E^2(\kappa)} = \frac{\sum_{\kappa=\kappa_1}^{\kappa_N}(N\kappa-1)^2}{N} \tag{31}$$

The Effective Sample Size (ESS) can then be formulated from which the decision to resample can be taken:

$$ESS = \frac{N}{1+CV} \tag{32}$$

Whether or not to resample can be intelligently decided by looking at the value of the *ESS*: when the ESS falls to a very low value in an iteration, the resampling scheme can be invoked. A common way to resample is to perform *selection with replacement* which implies unfit particles are rejected and their positions are filled with copies of fitter particles. This implies that fitter particles occupy the significant chunk of particles remaining in the posterior distribution. This result leads to the issue of *loss of diversity* when there are only a few particles that carry significant weights after an iteration.

Appendix C.

THE SWARM INTELLIGENCE PARADIGM

Swarm Intelligence (Beni, 2004 and Bonabeau et. al, 1999) [34-35] is a discipline motivating the design and analysis of new machine learning techniques and robotic systems and it studies large populations of simple agents which solve problems far too complex for an individual agent. These populations can also display the adaptability and robust to environmental change exhibited by biological agents. Some relevant terminologies are briefly elaborated on, in the following pages. For an extended reading on the subject, the reader is referred to any good text on bio-inspired computation, such as [36] from which the following is largely adopted:

Collective phenomena and the emergence of patterns

Collective phenomena are found in abundance in the biological world and render significant adaptive functions to individuals in a multi-agent system. These may manifest themselves as communication between local neighbours, coordinated movement in adherence to a set of low level rules that result in complex high-level group dynamics or simply niche formation where each sub swarm is an ecological model.

Self-organization

Self-organization indicates a process where structures at complex levels may be formed by accumulating local information exchange among simple agents. The resulting complex patterns

are called *emergent patterns* because these have implications far greater than that of the sum of its formative parts. Equilibrium in such systems rise out of an interplay between positive and negative feedback and this equilibrium state is equivalent to an attractor in dynamical systems theory. Systems will tend to return to it when perturbed. A self-organizing system can also manifest multiple states and chaotic trajectories. These systems can be described by sets of differential equations where the change of state is dependent on the state at the previous iteration (positive feedback) and a limiting factor (negative feedback) with magnitude inversely proportional to the magnitude of the state.

Aggregation

Aggregation is an example of self-organization that is best explained by positive and negative feedback mechanisms. In fish schools, a large number of individuals can swim in close formation that may rapidly change direction, reunite or disperse at will. The coordinated movement of the school gives it the appearance of a singular superorganism. Huth and Wissel (1992) [37] suggested a simple model of schooling based on both negative and positive feedback. A fish displays four behavioural reactions that depend on the position and orientation of other fish:

(1) If there are other fish in the immediate neighbourhood, then an individual fish of interest will move away to avoid collision *(Negative Feedback).*

(2) If there is another fish at an intermediate distance, the individual of interest will align its orientation.

(3) If there is a fish at a greater distance, the individual will tend to move towards it *(Positive Feedback).*

(4) If there is no fish in sight, the individual of interest will perform random search movements.

Clustering

Ant species are known to engage in clustering and sorting behavior. The probability that an ant picks up an object is proportional to the number of objects it has experienced within a short time window. Therefore, ants tend to pick up objects that isolated objects but do not remove objects that are found in clusters. Also, the probability that an ant deposits an object is proportional to the number of perceived objects in a short time window. Therefore, ants are likely to deposit an object near larger clusters of objects (Deneubourg et. al, 1991 and Theraulaz et. al, 2002). [38-39]

Nest Construction

Termites and wasps collectively build nests whose architectural complexities exceed the perceptual and cognitive abilities of individual agents. Models that seek to explain how such engineering achievements are accomplished without a seemingly evident plan rely mostly on *stigmergic communication* (Grassé, 1959) [40].

Foraging

Stigmergy can also improve the efficiency of collective foraging. Deneubourg et. al (1990) [41] showed that when pheromone laying ants are presented with two choices of a path with equal length between nest and food area, they choose the paths with equal probability initially. However they crowd one path soon enough and reject the other. This happens because the path with a stronger pheromone trail owing to a larger initial ant population visiting it will attract more ants and this creates a positive feedback loop.

Division of Labour

In many self-organizing models, all agents have the same set of behavioural rules and perform the same task. However, many insect societies also display division of labour and specialization where certain individuals are assigned specific tasks. Genetic factors such as polyethism (age-dependent specialization) and polymorphism (varying shape of body) impact the task allocation to an extent. Bonabeau et. al (1996) [42] proposed a *response threshold model* which states that an individual performs a task if the stimulus associated is greater than the individual's threshold.

REFERENCES

[1] Kass, M., Witkin, A. and Terzopoulos, D., “Snakes: Active Contour Models”, Int. J. Comput. Vis., vol. 1, no. 4, pp. 321-331, 1988.

[2] Hager, G.D., Belhumer, P.N., “Efficient Region Tracking with Parametric Models of Geometry and Illumination”, IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 10, Oct. 1998.

[3] Comaniciu, D., Ramesh, V. and Meer, P., “Kernel-based object tracking,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 5, pp. 564–577, Apr. 2003.

[4] Bray, M., Meier, E., Schraudolph, N., Van Gool, L., “Fast stochastic optimization for articulated structure tracking”, Image and Vision Computing vol. 25 no. 3 pp. 352– 364, 2007.

[5] Leung, A.P., Gong, S.P., “Optimizing Distribution-based Matching by Random Subsampling”, Proc. CVPR’07, pp. 1-8, 2007.

[6] Yang, C., Duraiswami, R., Davis, L., “Fast Multiple Object Tracking via a Hierarchical Particle Filter”, Proc. ICCV ’05, pp 212-219, 2005.

[7] Sun, B., Wang, B., Shi, Y., Gao, H., "Visual Tracking Using Quantum-Behaved Particle Swarm Optimization", Proc. of the 34th Chinese Control Conf, pp. 3844-3851, 2015.

[8] Hu, J., Fang, W., Ding. W, "Visual Tracking by Sequential Cellular Quantum-Behaved Particle Swarm Optimization", Bio-Inspired Computing - Theories and Applications (BIC-TA 2016), pp 86-94, 2016.

[9] Walia, G.S., Kapoor, R., “Intelligent video target tracking using an evolutionary particle filter based upon improved cuckoo search”, Expert Systems with Applications, vol. 41, issue 14, pp. 6315-6326, 2014.

[10] Gao, Ming-Liang., Shen, Jin., Yin, Li-Ju., Liu, Wei., Zou, Guo-Feng., Li, Hai-Tao., Fu, Gui-Xia., “A novel visual tracking method using bat algorithm”, Neurocomputing, Volume 177, pp. 612-619, 2016.

[11] Gao, Ming-Liang ., Li, Li-Li., Sun, Xian-Ming., Yin, Li-Ju., Li, Hai-Tao., Luo, Dai-Sheng., “Firefly algorithm (FA) based particle filter method for visual tracking”, Optik - International Journal for Light and Electron Optics, Volume 126, Issue 18, pp. 1705-1711, 2015.

[12] Gordon, N.J., Salmond, D.J., Smith, A.F.M., “Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation”, IEE Proc., F, Radar Signal Process. 140 (2), pp 107-113, 1993.

[13] Chang, C. and Ansari, R., “Kernel particle filter for visual tracking”, IEEE Signal Processing Letters, vol. 12, no. 3, pp. 242-245, 2005.

[14] Musso, C., Oudjane, N. and Gland, F., "Improving Regularised Particle Filters", Sequential Monte Carlo Methods in Practice, pp. 247-271, 2001.

[15] Green, P., "Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination", Biometrika, vol. 82, no. 4, pp. 711, 1995.

[16] Liu, J., Chen, R. and Wong, W., "Rejection Control and Sequential Importance Sampling", Journal of the American Statistical Association, vol. 93, no. 443, pp. 1022-1031, 1998.

[17] Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T., "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, IEEE Trans. Signal Process. 50(2) pp. 174-188, 2002.

[18] MacCormick, J., and Blake, A., "A probabilistic exclusion principle for tracking multiple objects", in Proc. Int. Conf. Comput. Vision, 1999, pp. 572–578.

[19] Radford, M.N., "Annealed importance sampling", Statistics and Computing, 11(2), pp. 125–139, 2001.

[20] Pitt, M.K., Shephard, N., "Filtering via simulation: Auxiliary particle filter", Journal of American Statistical Association, 94, pp. 590–599, 1999.

[21] Kennedy, J., Eberhart, R., "Particle swarm optimization.", Proc. IEEE Int. Conf. Neural Network, 1995.

[22] Van den Bergh, F., "An analysis of particle swarm optimizers."7, Ph.D. Thesis, University of Pretoria, November 2001.

[23] Solis, F.J., Wets, R. J-B., "Minimization by random search techniques.", Mathematics of Operations Research 6 pp. 19–30, 1981.

[24] Eberhart, R., "A discrete binary version of the particle swarm algorithm". In: Proceedings of 1997 conference systems man cybernetics, NJ: Piscataway; pp. 4104–8,1997.

[25] Sun, J., Xu,W.B., Feng, B., "A global search strategy of quantum-behaved particle swarm optimization.", Cybernetics and Intelligent Systems Proceedings of the 2004 IEEE Conference, pp. 111–116, 2004.

[26] Sun, J., Feng, B., Xu, W.B., "Particle swarm optimization with particles having quantum behavior.", IEEE Proceedings of Congress on Evolutionary Computation, pp. 325–331, 2004.

[27] Xi, M., Sun, J., Xu, W.B., “An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position”, Applied Mathematics and Computation, pp. 751-759, Nov 2008.

[28] Clerc, M., Kennedy, J., “The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space”, IEEE Transactions on Evolutionary Computation 6(1), pp. 58–73, 2002.

[29] Liu. J., Sun. J., Xu. W., “Improving Quantum-Behaved Particle Swarm Optimization by Simulated Annealing”, Computational Intelligence and Bioinformatics, LNBI 4115, pp. 130 – 136, Springer-Verlag Berlin Heidelberg 2006.

[30] Metropolis, N. et al, “Equation of state calculations by fast computing machines,” The Journal of Chemical Physics, vol. 21, no. 6, pp. 1087–1092, 1953.

[31] Fisher, R., “Caviar case scenarios”, http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/.

[32] Mian, A., "Realtime Visual Tracking of Aircrafts'', Digital Image Computing:Techniques and Applications (DICTA), pp. 351-356, 2008.

[33] Kirkpatrick, S., Gelatt, C., Vecci, M., “Optimization by Simulated Annealing”, Science 220:671-680, 1983.

[34] Bonabeau, E., Dorigo, M., and Theraulaz, G., “Swarm Intelligence: From Natural to Artificial Systems”, Oxford University Press, New York, 1999.

[35] Beni, G., "From swarm intelligence to swarm robotics. In. Sahin, E. and Spears", W. M., Editors, Proceedings of the Swarm Robotics Workshop, pages 1–9. Springer Verlag, Heidelberg, Germany, 2004.

[36] Floreano, D., Mattiussi, C., “Bio-inspired Artificial Intelligence: Theories, Methods, and Technologies”, MIT press, 2008

[37] Huth, A., Wissel, C., “The simulation of the movement of fish schools, Journal of Theoretical Biology”, 156:365–385, 1992.

[38] Deneubourg, J.-L., Goss, S., Franks, N. R., Sendova-Franks, A., Detrain, C., and Chretien, L., “The dynamics of collective sorting: Robot-like ant and ant-like robot”, In Meyer, J.-A. and Wilson, S. W., editors, Proc. of the First International Conference on Simulation of Adaptive Behavior: From Animals to Animats, pages 356–365.MIT Press, Cambridge, MA, 1991.

[39] Theraulaz, G., Bonabeau, E., Nicolis, S., Solé, R. V., Fourcassié, V., Blanco, S., Fournier, R., Joly, J.-L., Fernandez, P., Grimal, A., Dalle, P., and Deneubourg, J.-L. ”Spatial patterns in ant colonies”. Proceedings of the National Academy of Sciences USA, 99:9645–9649, 2002.

[40] Grassé, P.-P., “La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes natalensis et Cubitermes sp. la théorie de la stigmergie: Essai d’interprétation du comportement des termites constructeurs. Insectes Sociaux, 6:41–83, 1959.

[41] Deneubourg, J.-L., Gregoire, J. C., and Le Fort, E., “Kinetics of the larval gregarious behaviour in the bark beetle”, Dendroctonus micans. Journal of Insect Behavior, 3:169–182, 1990.

[42] Bonabeau, E., Theraulaz, G., and Deneubourg, J.-L., “Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies”, Proceedings of the Royal Society of London B, 263:1565–1569, 1996.